

Genhol - a utility that generates user-defined moving holiday regressors for X-13ARIMA-SEATS

U. S. Census Bureau
August 28, 2020

- [Running Genhol](#)
 - [Notes on centering holiday regressors](#)
 - [Examples](#)
 - [Generating an Easter holiday regressor](#)
 - [Generating a 2-part Easter holiday regressor](#)
 - [Generating a stock Easter holiday regressor](#)
 - [Generating a set of regressors for more than one holiday](#)
 - [Support](#)
 - [References](#)
-

The Genhol utility creates holiday regressors using the same procedure as X-13ARIMA-SEATS uses to create regressors for the U. S. holidays of Easter and Labor Day. Separate regressors can be generated for the effect in intervals before and after the holiday, as well as an intermediate effect around the holiday. More than one holiday can be specified in a single input file, and the user can center these regressors using either their overall means or their calendar-month means (calendar-quarter means in the case of regressors for modeling quarterly data or calendar-bimonthly means in the case of regressors for modeling bimonthly data).

X-13ARIMA-SEATS commands can be generated to allow the user to easily use these regressors in a regARIMA model for an X-13ARIMA-SEATS run. For more details on specific options for user-defined regressors, go to the section on the `regression spec` in U. S. Census (2020).

Holiday effects for flow and stock time series can be generated with this software.

Files for some holidays are available from this website. For holidays whose dates are not available, the user must supply an ASCII (text) file of dates.

NOTE: This program was written with the Icon programming language, with special routines provided by the Icon program library. For more information on Icon, see Griswold and Griswold (1996).

The source code for this program is available upon request to the person listed under **Support**.

Running Genhol

To run Genhol, enter the following command in a (DOS) command prompt:

```
genhol infile
```

where `infile` is a file that contains options for Genhol. The general syntax of this file is similar to that of an X-13ARIMA-SEATS input specification file, see below:

```
spec1{
    argument = value
}
spec2{
    argument = value
}
```

There are two types of "specs": a `global` spec (to specify global options) and `holiday` specs (to specify options for each of the holidays for which user-defined regressors are to be generated). The global spec should appear first in the input file.

The arguments that can be specified in the `global` spec are listed in the table below.

Possible arguments for the `global` spec

Argument	Description
<code>numhol = n</code>	Denotes the number of separate holidays to be specified by the user later in this file. The default is 1.
<code>outfile = name</code>	Filename (with extension) where the holiday regression variables will be stored. This argument must be specified; there is no default.
<code>outspec = name</code>	Filename (with extension) where a regression spec will be stored that uses the user-defined regression

	variables generated by this program. If not specified, this file will not be generated.
<code>first = nnnn</code>	Denotes the first year that will be saved in the output file. The default is the first year common to all the data files specified by the <code>infile</code> argument of subsequent holiday specs.
<code>last = nnnn</code>	Denotes the last year that will be saved in the output file. The default is the final year common to all the data files specified by the <code>infile</code> argument of subsequent holiday specs.
<code>firstmean = nnnn</code>	Denotes the first year used in the calculation of the means of the holiday regressors. The default is the first year common to all the data files specified by the <code>infile</code> argument of subsequent holiday specs.
<code>lastmean = nnnn</code>	Denotes the last year used in the calculation of the means of the holiday regressors. The default is the final year common to all the data files specified by the <code>infile</code> argument of subsequent holiday specs.
<code>period = n</code>	Denotes the seasonal period assumed for the holiday regression variables being generated and stored. Only three values are allowed for this argument: <code>period=12</code> (to generate regressors for monthly time series), <code>period=6</code> (to generate regressors for bimonthly series), and <code>period=4</code> (to generate regressors for quarterly series). The default is 12.
<code>type = string</code>	Denotes the type of variable that will be produced. If <code>type = count</code> , the count of the number of days found in the holiday window for a given observation is output. If <code>type = ratio</code> , the ratio of the number of days in the holiday window for a given observation is output. The default is <code>type = ratio</code> .
<code>stockday = n</code>	Denotes the stock day for an inventory series, that is, the day when inventory is taken for a given month. This can only be specified for monthly series, and can only take the values of 1 to 31 (ie, $1 \leq \text{stockday} \leq 31$). An end-of-month inventory regressor is generated when <code>stockday = 31</code> . The default is to assume that the holiday generated will be for a flow series rather than an inventory series.

<code>usergroup = string</code>	Determines whether user-defined regression groups will be specified in the <code>regression spec</code> stored in <code>outspec</code> . If <code>usergroup = yes</code> , the regressors generated for each holiday will be put in separate user-defined regression groups. This can only be specified if more than one holiday is specified (ie, <code>numhol > 0</code>). If <code>usergroup = no</code> , no user-defined regression groups will be specified. The default is <code>usergroup = no</code> .
<code>chi2test = string</code>	Determines whether a chi-squared test of the user-defined regression groups will be specified in the <code>regression spec</code> stored in <code>outspec</code> . If <code>chi2test = yes</code> , a chi-squared test of the user-defined regression groups will be specified in the <code>regression spec</code> stored in <code>outspec</code> . This can only be specified if more than one holiday is specified (ie, <code>numhol > 0</code>) and if <code>usergroup = yes</code> . If <code>usergroup = no</code> , no user-defined regression groups will be specified. The default is <code>usergroup = no</code> .

Note that the file specified in `outfile` is where the regression variables will be stored; the file specified in `outspec` is where arguments for the X-13ARIMA-SEATS `regression spec` will be stored that can read and process the regression variables stored in `outfile`.

After the `global spec`, there should be as many `holiday` specs as specified in the `numhol` argument of the `global spec`. These specs should be numbered (ie, `holiday1`, `holiday2`, etc.).

The arguments for the `holiday` specs are given below.

Possible arguments for the `holiday spec(s)`

Argument	Description
<code>name = string</code>	Name of the holiday. This name should not have any spaces (this name will be used in the regression spec generated by the program to distinguish between different holiday regressors). Default is the spec file name (<code>holiday1</code> , <code>holiday2</code> , etc.).

<code>infile = name</code>	Specifies the filename (with extension) where the holiday dates used to generate the holiday regressors are stored. Each date consists of a month, day, and four digit year separated by white space or tabs (example: 1 1 1990 for January 1, 1990), and each date should appear on its own line. This is a required entry.
<code>begbefore = n</code>	Denotes the position relative to the holiday of the beginning of the window used to generate the before-holiday regressor. This value should be negative, and less than or equal to the value for the <code>endbefore</code> argument. The minimum value that can be specified is -42.
<code>endbefore = n</code>	Denotes the position relative to the holiday of the end of the window used to generate the before-holiday regressor. This value should be negative.
<code>begafter = n</code>	Denotes the position relative to the holiday of the beginning of the window used to generate the after-holiday regressor. Since this effect occurs after the holiday, the value should be non-negative.
<code>endafter = n</code>	Denotes the position relative to the holiday of the end of the window used to generate the after-holiday regressor. This value should be positive, and greater than or equal to the value for the <code>begafter</code> argument. The maximum value that can be specified is 49.
<code>center = string</code>	Specifies the removal of the (sample) mean or the seasonal means from the user-defined regression variables. If <code>center = mean</code> , the mean of the user-defined holiday regressor is subtracted from the regressor. If <code>center = calendar</code> , means for each calendar month (or quarter or bimonth) are subtracted from each of the user-defined holiday regressors. If this argument is not specified, the user-defined regressors are not modified.

In general, here are some guidelines for choosing which options to specify, depending on the type of effect you wish to estimate.

- To specify a regressor to estimate an effect before a holiday, the user must specify both the `begbefore` and `endbefore` arguments of the `holiday spec`.
- In general, $-42 \leq \text{begbefore} \leq \text{endbefore} < 0$.
- To specify a regressor to estimate an effect after a holiday, the user must specify both the `begafter` and `endafter` arguments.
- In general, $0 < \text{begafter} \leq \text{endafter} \leq 49$.
- An interim regressor, using a window around a given holiday, will also be generated when the user specifies values for both the `endbefore` and `begafter` arguments. This effect will use a window that starts one position after the position given by `endbefore` and ends one position before the position given by `begafter`.
- If only the interim regressor is desired, the user should specify just the `begafter` and `endbefore` arguments.

[Back to the Top](#)

Notes on centering holiday regressors

For simplicity, only regressors for monthly data are discussed. The quarterly and bimonthly cases are analogous.

Before centering, each regressor is a proportionality regressor whose value in any month is the proportion of the days of the specified interval that belong to the month. For example, if the interval is 10 days long and, in a certain year, the first two days of the interval fall in January, then the value of the uncentered regressor is 2/10 in January and 8/10 in February that year. Its value in March-December is 0.

Centering of the regressors is done to keep the yearly totals of the series obtained by removing the estimated holiday effects approximately equal to the yearly totals of the unadjusted data. If centering is not done, these two totals will differ by approximately the same amount each year, and user will conclude that combined seasonal and holiday adjustment is producing a biased estimate of the level of the observed series. (The bias is a crude estimate of what the data would be

like if there were no holiday.) The type of holiday effect determines the type of centering.

Centering by removing the calendar-month means is appropriate for holidays like Easter or the Chinese New Year whose regressors are always zero in some calendar months because the holiday can occur only in a few calendar months. The calendar-month-centered regressors will be zero in exactly the same months as the regressor, and no adjustment will be done to data from these months. Also, holiday adjustment does remove fixed seasonal, with the result that all such effects are included in the seasonal factors produced by the seasonal adjustment procedure.

Centering by removing the overall mean is attractive for holiday periods like Ramadan that move through all of the calendar months over time. With mean centering, the centered regressor's adjustment for every month outside the specified interval in a given year always has the same value and adjustment for the holiday effect has a usually negligible effect on the general level of the series.

[Back to the Top](#)

Examples

Generating an Easter holiday regressor

This example will create a file containing user-defined regressors for Easter. This file will have values very close to those of the default regressor generated by X-13ARIMA-SEATS for Easter with a window length of 8 (`easter[8]`). A sample input file (`easter8.inp`) is given below:

```
global{
  numhol = 1
  outfile = "easter8.dat"
  outspec = "easter8.reg"
  first = 1990
  last = 2015
  period = 12
}
```

```

holiday1{
    name = easter8
    begbefore = -8
    endbefore = -1
    infile = "easter500.txt"
    center = calendar
}

```

Note that the dates of Easter holiday is provided in the file `easter500.txt`. The first 15 lines of the file are given below:

```

4      2      1600
4     22      1601
4      7      1602
3     30      1603
4     18      1604
4     10      1605
3     26      1606
4     15      1607
4      6      1608
4     19      1609
4     11      1610
4      3      1611
4     22      1612
4      7      1613
3     30      1614
4     19      1615

```

To run this file, enter

```
genhol easter8.inp
```

This will generate one holiday regressor for the Easter holiday, stored in `easter8.dat`. The regression variable is stored in X-13ARIMA-SEATS's `datevalue` format; the first 18 lines of the file are given below:

```

1990 1  0.0
1990 2  0.0
1990 3 -0.382
1990 4  0.382
1990 5  0.0
1990 6  0.0
1990 7  0.0
1990 8  0.0
1990 9  0.0
1990 10 0.0
1990 11 0.0
1990 12 0.0
1991 1  0.0
1991 2  0.0

```



```

1991 3  0.618
1991 4 -0.618
1991 5  0.0
1991 6  0.0

```

X-13ARIMA-SEATS commands to incorporate these regressors into a regARIMA model will also be stored into a file named `easter8.reg`. The contents of this file are given below:

```

regression{
  user=(
    Beforeeaster8
  )
  file="easter8_500.dat"
  format="datevalue"
  start=1980.1
  usertype=holiday
}

```

[Back to the Top](#)

Generating a 2-part Easter holiday regressor

Zhang, McLaren, and Leung (2003) developed an alternate Easter regressor for Australian series at the Australian Bureau of Statistics (henceforth ABS). Their model consisted of two regressors, one which modeled an assumed increase in the level of the series before Good Friday, and another to handle the period between Good Friday and Easter Monday, a national holiday in Australia, where a decrease in level is expected.

This example will show how to create user-defined regressors for the ABS Easter regressor. A sample input file (`abseaster.inp`) is given below:

```

global{
  numhol = 1
  outfile = "abseaster.dat"
  outspec = "abseaster.reg"
  first = 1990
  last = 2015
  period = 12
}
holiday1{
  name = abseaster

```

```

    begbefore = -8
    endbefore = -3
    begafter = 2
    infile = "easter500.txt"
    center = calendar
}

```

To run this file, enter

```
genhol abseaster.inp
```

This will generate two (2) holiday regressors in the file `abseaster.dat` - a regressor for the period before Good Friday, and another regressor for the period between Good Friday and Easter Monday, inclusive.

X-13ARIMA-SEATS commands to incorporate these regressors into a `regARIMA` model will also be stored as a file named `abseaster.reg`. The contents of this file are given below:

```

regression{
  user=(
    Beforeabseaster    Betweenabseaster
  )
  file="abseaster.dat"
  format="datevalue"
  start=1990.1
  usertype=holiday
}

```

[Back to the Top](#)

Generating a stock Easter holiday regressor

This utility uses the method outlined in Findley (2009) to generate stock holiday regressors. To generate an end-of-month stock Easter regressor approximately like that specified in X-13ARIMA-SEATS as `easterstock[8]` (see Monsell (2009) for more details), an input file like the following is needed:

```

global{
  numhol = 1
  outfile = "stockeaster8.dat"
  outspec = "stockeaster8.reg"
}

```

```

        first = 1990
        last = 2015
        period = 12
        stockday = 31
    }
    holiday1{
        name = stockeaster8
        begbefore = -8
        endbefore = -1
        infile = "easter500.txt"
        center = calendar
    }

```

To generate a stock Easter regressor for another sample day, such as the 28th day of the month, an input file like the following is needed:

```

global{
    numhol = 1
    outfile = "stockeaster8d28.dat"
    outspec = "stockeaster8d28.reg"
    first = 1990
    last = 2015
    period = 12
    stockday = 28
}
holiday1{
    name = stockeaster8d28
    begbefore = -8
    endbefore = -1
    infile = "easter500.txt"
    center = calendar
}

```

[Back to the Top](#)

Generating a set of regressors for more than one holiday

This example will create a file containing user-defined regressors for Chinese New Year and Idul Fitri, an Islamic holiday that marks the end of Ramadan. A sample input file (indhhol.inp) is given below:

```

global{
    numhol = 2
    outfile = "indhhol.dat"
    outspec = "indhhol.reg"
    usergroup = yes
    chi2test = yes
}

```

```

}
holiday1{
    name = CNY
    begbefore = -10
    endbefore = -1
    begafter = 5
    endafter = 14
    infile = "cnyInd.dat"
    center = calendar
}
holiday2{
    name = IdulFitri
    begbefore = -42
    endbefore = -21
    begafter = 1
    endafter = 7
    infile = "IFHolInd.dat"
    center = mean
}

```

To run this file, enter

```
genhol indhol.inp
```

This will generate six (6) holiday regressors in the file `indhol.reg` - three Chinese New Year regressors (for before, after and intermediate holiday effects), and Idul Fitri regressors (for before, after and intermediate holiday effects). X-13ARIMA-SEATS commands to incorporate these regressors into a regARIMA model will be stored as a file named `indhol.reg`. The contents of this file are given below:

```

regression{
    user=(
        BeforeCNY    BetweenCNY    AfterCNY
        BeforeIdulFitri    BetweenIdulFitri
        AfterIdulFitri
    )
    file="indhol.dat"
    format="datevalue"
    start=1970.1
    usertype=(
        holiday    holiday    holiday
        holiday2    holiday2    holiday2
    )
    chi2test=yes
}

```

[Back to the Top](#)

Support

email: x12@census.gov

[Back to the Top](#)

References

Findley, D. F. (2009), **Stock Series Holiday Regressors Generated By Flow Series Holiday Regressors** [PDF], Statistical Research Division Research Report Series (Statistics No. 2009-04). U.S. Census Bureau [Online]. Available: <https://www.census.gov/content/dam/Census/library/working-papers/2009/adrm/rrs2009-04.pdf> [2020, September 1].

Griswold, R. E. and Griswold, M. T. (1996) **The Icon Programming Language, Third Edition** Peer-to-Peer Communications [Online]. Available: <http://www.cs.arizona.edu/icon/lb3.htm> [2009, July 24].

Monsell, B. C. (2009), **Update on the Development of X-13ARIMA-SEATS**, Proceedings of the 2009 Joint Statistical Meetings. Available: <https://www.census.gov/content/dam/Census/library/working-papers/2009/adrm/jsm09bcm.pdf> [2020, September 1].

U.S. Census Bureau (2020). **X-13ARIMA-SEATS Reference Manual, Version 1.1** [PDF] U.S. Census Bureau [Online]. Available: <https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf> [2020, September 1].

Zhang, X., McLaren, C. H., and Leung, C. C. S. (2003), **An Easter proximity effect: Modelling and adjustment**, *Australian and New Zealand Journal of Statistics*, 43, 269–280.

[Back to the Top](#)